

# TP 2 - XPath et XQuery

## Partie 1 :

### 1 - XPath : Films

- 1) //TITRE
- 2) //TITRE[ANNEE/text()='1990]
- 3) //RESUME[preceding-sibling::TITRE/text()='Alien']
- 4) //FILM[last()]
- 5) (/FILM[ANNEE=1990]/self::\*)[last()]
- 6) (/FILM[ANNEE=1990])[position()=count(/FILM[ANNEE=1990])]
- 7) //FILM[RESUME]/TITRE
- 8) //FILM[not(RESUME)]/TITRE
- 9) //ARTISTE[@id=//FILM[TITRE='Vertigo']/ROLE/@idref]/ACTNOM
- 10) //ARTISTE[@id=//FILM[TITRE='Vertigo']/MES/@idref]/ACTNOM
- 11) //FILM[MES/@idref=//FILM[TITRE='Vertigo']/MES/@idref]
- 12) //FILM/TITRE[contains(.,'V')]
- 13) //FILM[count(/FILM[TITRE='Shining']/preceding-sibling::\*)]/TITRE
- 14) //FILM/TITRE[../ROLE/@idref=//FILMS/ARTISTE/@id[(../ACTNOM='Willis')and(../ACTPNOM='Bruce')]]
- 15) //FILM/TITRE[(../ROLE/@idref=//FILMS/ARTISTE/@id[(../ACTNOM='Willis')and(../ACTPNOM='Bruce')])and(../ROLE/@idref=//FILMS/ARTISTE/@id[(../ACTNOM='Mendes')and(../ACTPNOM='Sam')])]
- 16) //FILM[TITRE='Reservoir dogs']/ROLES/ROLE[@idref=//ARTISTE/@id[(../ACTNOM='Keitel')and(../ACTPNOM='Harvey')]]
- 17) //ARTISTE[@id=//FILM[TITRE='Reservoir dogs']/ROLE/@idref]/ACTNOM[text()!='Keitel']
- 18) descendant::node()[count(descendant::node())=3]
- 19) descendant::node\[ \ ][contains\(name\[ \ ],"TI" \ )]

### 2 - XPath : Recettes

- 1) //recette/@nom
- 2) //recette[@nomCourt='Chiffonnade']/ingredient
- 3) /child::recettes/child::recette/@nom[parent::recette/child::materiel/child::ingredient/child::text()='persil haché']
- 4) //recette/@nom[../ingredient/text()='persil haché']
- 5) //recette/@nom[count(../ingredient)>2 and ../ingredient/text()='huile d'olive']
- 6) //recette/@nom[../ingredient[position()=3] and ../ingredient/text()='huile d'olive']
- 7) //recette[last()]

### 3 - XPath : Trains

- 1) `//train[self::*//bar]`
- 2) `//usager[@id=//resa/@id]/@nom`
- 3) `//train[.//resa[position()>=2]]`
- 4) `//resa/@id[count(.//resa/@id[.=premierid])=2]` Impossible dans XPath car on oublie la valeur du premier id dans la deuxième imbrication.

## **Extra**

- 1) `//voiture[not ( //voiture[ count(.//resa) > count(premierVoiture//resa) ) ] ]` On a une double imbrication.

On ne peut pas comparer le nombre de passagers dans une voiture avec le nombre de passagers dans toutes les autres voitures car on ne peut pas faire le lien entre les passagers de deux voitures car il y a plusieurs imbrications.

- 2) `//usager[not(@id=//resa/@id)]`
- 3) Impossible car il faudrait faire : `"//resa[string-length(@id1)>(//resa[string-length(@id2)] and id1 != id2)]"`

Or il est impossible de faire le lien entre les deux ids.

## **Partie 2 :**

### **2 - XQuery sur les Tweets**

Voici notre document xml :

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE twitter [
<ELEMENT twitter (utilisateur* | ( utilisateur+, tweet*)) >

<ELEMENT tweet (post, date, coordonnees?, os?, cptRetweet)>
<!ATTLIST tweet idTweet ID #REQUIRED>
<!ATTLIST tweet idAuthor IDREF #REQUIRED>
<!ATTLIST tweet reponse IDREF #IMPLIED>    <!-- référence vers le tweet précédent-->
<!ATTLIST tweet allowAnswers (true|false) #REQUIRED>

<ELEMENT post (retweet*|message|video|image)+>
<!ELEMENT retweet (#PCDATA)>
<!ATTLIST retweet refTweet IDREF #REQUIRED>
<ELEMENT message ((text|hashtag|refUser)+, size, font, color, language)>
<!ELEMENT text (#PCDATA)>
<!ELEMENT hashtag (#PCDATA)>
<!ELEMENT refUser (#PCDATA)>
<!ELEMENT size (#PCDATA)>
<!ELEMENT font (#PCDATA)>
<!ELEMENT color (#PCDATA)>
<!ELEMENT language (#PCDATA)>
```

[illegible]

```

<post>
  <message>
    <text>Le match de foot était aussi intéressant que ma vie</text>
    <hashtag>#I&lt;3XML</hashtag>
    <size>14</size>
    <font>Arial</font>
    <color>Red</color>
    <language>French</language>
  </message>
</post>
<date>
  <secondes>1467111111</secondes>
  <fuseauHoraire>UTC+1</fuseauHoraire>
</date>
<coordonnees>
  <gps>
    <longitude>3.8767159999999876</longitude>
    <latitude>43.610769</latitude>
  </gps>
  <ville>Montpellier</ville>
  <pays>France</pays>
</coordonnees>
<os>Windows XP</os>
<cptRetweet>3</cptRetweet>
</tweet>
<tweet idTweet="x2008" idAuthor="x0001" allowAnswers="true">
  <post>
    <retweet refTweet="x2001"/>
    <message>
      <text>Le match de foot était aussi intéressant que ma vie</text>
      <size>14</size>
      <font>Arial</font>
      <color>Red</color>
      <language>French</language>
    </message>
  </post>
  <date>
    <secondes>1467222222</secondes>
    <fuseauHoraire>UTC+1</fuseauHoraire>
  </date>
  <coordonnees>
    <gps>
      <longitude>3.8767159999999876</longitude>
      <latitude>43.610769</latitude>
    </gps>
    <ville>Montpellier</ville>
    <pays>France</pays>
  </coordonnees>
  <os>Windows XP</os>
  <cptRetweet>0</cptRetweet>
</tweet>

```

```
<tweet idTweet="x2007" idAuthor="x0001" allowAnswers="true">
  <post>
    <retweet refTweet="x2001"/>
    <message>
      <text>Le match de foot était aussi intéressant que ma vie</text>
      <refUser>x0002</refUser>
      <refUser>x0003</refUser>
      <size>14</size>
      <font>Arial</font>
      <color>Red</color>
      <language>French</language>
    </message>
  </post>
  <date>
    <secondes>1467333333</secondes>
    <fuseauHoraire>UTC+1</fuseauHoraire>
  </date>
  <coordonnees>
    <gps>
      <longitude>3.8767159999999876</longitude>
      <latitude>43.610769</latitude>
    </gps>
    <ville>Montpellier</ville>
    <pays>France</pays>
  </coordonnees>
  <os>Windows XP</os>
  <cptRetweet>0</cptRetweet>
</tweet>
```

```
<tweet idTweet="x2006" idAuthor="x0001" allowAnswers="true">
  <post>
    <retweet refTweet="x2001"/>
    <message>
      <text>Le match de foot était aussi intéressant que ma vie</text>
      <size>14</size>
      <font>Arial</font>
      <color>Red</color>
      <language>French</language>
    </message>
  </post>
  <date>
    <secondes>1467444444</secondes>
    <fuseauHoraire>UTC+1</fuseauHoraire>
  </date>
  <coordonnees>
    <gps>
      <longitude>3.8767159999999876</longitude>
      <latitude>43.610769</latitude>
    </gps>
    <ville>Montpellier</ville>
    <pays>France</pays>
  </coordonnees>
```

```
<os>Windows XP</os>
<cptRetweet>0</cptRetweet>
</tweet>
<tweet idTweet="x2002" idAuthor="x0001" allowAnswers="false">
  <post>
    <message>
      <text>Malheureusement aujourd'hui c'est repas chez la belle mère... Elle va encore cracher
dans ma soupe.</text>
      <size>14</size>
      <font>Arial</font>
      <color>Black</color>
      <language>French</language>
    </message>
  </post>
  <date>
    <secondes>1468111111</secondes>
    <fuseauHoraire>UTC+1</fuseauHoraire>
  </date>
  <coordonnees>
    <gps>
      <longitude>3.8767159999999876</longitude>
      <latitude>43.610769</latitude>
    </gps>
    <ville>Montpellier</ville>
    <pays>France</pays>
  </coordonnees>
  <os>iOS 9.1.3</os>
  <cptRetweet>0</cptRetweet>
</tweet>
<tweet idTweet="x2003" idAuthor="x0002" allowAnswers="false">
  <post>
    <message>
      <text>Regardez cette vidéo de chats! Trop chou!</text>
      <size>12</size>
      <font>Calibri</font>
      <color>Black</color>
      <language>French</language>
    </message>
    <video>
      <url>https://www.google.fr/url?sas</url>
    </video>
  </post>
  <date>
    <secondes>1470472534</secondes>
    <fuseauHoraire>UTC+1</fuseauHoraire>
  </date>
  <coordonnees>
    <gps>
      <longitude>3.8767159999999876</longitude>
      <latitude>43.610769</latitude>
    </gps>
```

```

    <ville>Montpellier</ville>
    <pays>France</pays>
  </coordonnees>
  <os>iOS 9.1.3</os>
  <cptRetweet>0</cptRetweet>
</tweet>
<tweet idTweet="x2004" idAuthor="x0002" allowAnswers="false">
  <post>
    <message>
      <text>Je veux la même licrone!</text>
      <size>12</size>
      <font>Calibri</font>
      <color>Black</color>
      <language>French</language>
    </message>
    <image>
      <url>http://www.republ33k.fr/wp-content/uploads/2013/05/6784492-cute-pink-unicorn.jpg</url>
    </image>
  </post>
  <date>
    <secondes>1470501334</secondes>
    <fuseauHoraire>UTC+1</fuseauHoraire>
  </date>
  <coordonnees>
    <gps>
      <longitude>3.8767159999999876</longitude>
      <latitude>43.610769</latitude>
    </gps>
    <ville>Montpellier</ville>
    <pays>France</pays>
  </coordonnees>
  <os>iOS 9.1.3</os>
  <cptRetweet>0</cptRetweet>
</tweet>
<tweet idTweet="x2005" idAuthor="x0003" allowAnswers="false">
  <post>
    <message>
      <text>Mon premier tweet</text>
      <size>18</size>
      <font>Comic sans MS</font>
      <color>Pink</color>
      <language>French</language>
    </message>
  </post>
  <date>
    <secondes>1473179734</secondes>
    <fuseauHoraire>UTC+1</fuseauHoraire>
  </date>
  <coordonnees>
    <gps>
      <longitude>5.4474270000000615</longitude>

```

```

        <latitude>43.529742</latitude>
    </gps>
    <ville>Aix-en-Provence</ville>
    <pays>France</pays>
</coordonnees>
<os>iOS 9.1.3</os>
<cptRetweet>0</cptRetweet>
</tweet>
</twitter>

```

1)

```
let $x:= doc('twitter.xml')
```

```

for $y in $x//utilisateur, $z in $x//tweet
    where $y/@idAut = $z/@idAuthor
    return
    element
    {"result"}
    {
        element
        {"tweet"}
        {$z}
        ,
        element
        {"auteur"}
        {$y}
    }

```

2)

```
let $x:= doc('twitter.xml')
```

```

for $y in $x//utilisateur
    return
    element
    {"result"}
    {
        attribute
        {"idAut"}
        {$y//@idAut}
        ,
        element
        {"date"}
        {
            for $z in $x//tweet
                where $y/@idAut = $z/@idAuthor
                return $z/date
        }
    }

```

3)



```
let $x:= doc('twitter.xml')
```

```
for $y in $x//utilisateur
let $z := $x//tweet[./@idAuthor = $y/@idAut ] return
if ($z/cptRetweet >= 2) then
    $y
else
    ()
```

4)

```
let $x:= doc('twitter.xml')
```

```
for $y in $x//tweet
return
element
{"tweet"}
{
    attribute
    {"id"}
    {$y/@idTweet}
    ,
    element
    {"contenu"}
    {$y/post/message/text}
    ,
    if ($y//cptRetweet>=2) then
    let $z := $x//tweet[./retweet[@refTweet=$y/@idTweet]] return
    ($z[1]//date, $z[2]//date)
    else
    element
    {"nonRetwitted"}
    {}
}
```

5)

```
let $x:= doc('twitter.xml')
```

```
for $y in $x//utilisateur
order by $y/nom/text()
return $y
```

6)

```
let $x:= doc('twitter.xml')
```

```
for $y in $x//tweet
where $y/post/message/hashtag/text() = "#I<3XML"
return $y
```

7)

```
let $x := doc("twitter.xml")
```

```
for $y in $x//tweet return
```

```
if (not($y//secondes<$x//secondes)) then
```

```
    element
```

```
        {"plusRecentTweet"}
```

```
        {($y/@*, $y/node())}
```

```
else if (not($y//secondes>$x//secondes)) then
```

```
    element
```

```
        {"plusAncientTweet"}
```

```
        {($y/@*, $y/node())}
```

```
else
```

```
    ()
```

8)

```
let $x := doc("twitter.xml")
```

```
for $y in $x//tweet return
```

```
if (not($y//hashtag)) then
```

```
    ()
```

```
else
```

```
    element
```

```
        {"tweetHashtag"}
```

```
        {($y, $y//hashtag)}
```

9)

```
let $x := doc("twitter.xml")
```

```
for $y in $x//tweet return
```

```
if (not($y//refUser)) then
```

```
    ()
```

```
else
```

```
    element
```

```
        {"assoTweetRefUser"}
```

```
        {($y,
```

```
        element
```

```
            {"utilisateursReferences"}
```

```
            {$x//utilisateur[@idAut=$y//refUser]}
```

```
        )}
```

10)

```

declare function local:retwittePar($n) {
  let $x := doc('twitter.xml')

  let $y := $x//utilisateur[@idAut=$x//tweet[.//retweet/@refTweet=$n/@idTweet]/@idAuthor]
  return
  element
  {"retweeted"}
  {($n,
    if ($y) then
      element
      {"retweetedBy"}
      {$y}
    else
      element
      {"nonRetweeted"}
      {}
  )}
};

let $x := doc('twitter.xml')
for $y in $x//tweet return
local:retwittePar($y)

```

### **3 - Génération de pages HTML via XQuery**

```

let $x :=
doc('http://opendata.montpelliernumerique.fr/datastore/VilleMTP_MTP_ZATAntigone_2011.xml')
return
element {"html"}
{
  element {"head"}
  {
    element {"title"}
    {
      "Évènements culturels à Montpellier"
    }
  },
  element {"meta"}
  {
    attribute {"charset"}
    {
      "utf-8"
    }
  }
},
element {"body"}
{
  element {"h1"}

```

```

{
  "Évènements culturels à Montpellier"
}
,
element {"ul"}
{
  element {"li"}
  {
    attribute {"id"}
    {
      "byDate"
    }
  },
  element {"h2"}
  {
    "Par dates"
  },
  for $y in $x//spectacle
  order by $y//date return (
    element {"dl"}
    {
      element {"dt"}
      {
        "Date"
      },
      element {"dd"}
      {
        $y//date/text()
      },
      element {"dt"}
      {
        "Titre"
      },
      element {"dd"}
      {
        concat($y/title, if($y/author/text()) then
        concat(" par ", $y/author)
        else
        ())
      },
      element {"dt"}
      {
        "Sous-titre"
      },
      element {"dd"}

```

```

{
  if($y/subtitle/text()) then
    $y/subtitle/text()
  else
    ()
}
,
element {"dt"}
{
  "Lieu"
}
,
element {"dd"}
{
  concat("Latitude : ", $y//lat)
  ,
  concat("Longitude : ", $y//lng)
}
,
element {"br"}
{
}
}
)
}
,
element {"li"}
{
  attribute {"id"}
  {
    "byTitle"
  }
,
element {"dl"}
{
  element {"h2"}
  {
    "Par titre"
  }
,
  for $y in $x//spectacle
  order by $y//title return
  (
    element {"dt"}
    {
      "Titre"
    }
    ,
    element {"dd"}
    {
      concat($y/title, if($y/author/text()) then

```

```

        concat(" par ", $y/author)
    else
    ()
}
,
element {"dt"}
{
    "Sous-titre"
}
,
element {"dd"}
{
    if($y/subtitle/text()) then
        $y/subtitle/text()
    else
    ()
}
,
element {"dt"}
{
    "Date"
}
,
element {"dd"}
{
    $y/date/text()
}
,
element {"dt"}
{
    "Lieu"
}
,
element {"dd"}
{
    concat("Latitude : ", $y//lat)
    ,
    concat("Longitude : ", $y//lng)
}
,
element {"br"}
{
}
)
}
}
,
element {"li"}
{
    attribute {"id"}
{

```

```

    "byAuthor"
  }
  ,
  element {"h2"}
  {
    "Par auteurs"
  }
  ,
  for $y in $x//spectacle
  order by $y//author return (
    element {"dl"}
    {
      element {"dt"}
      {
        "Auteur"
      }
      ,
      element {"dd"}
      {
        if($y//author/text()) then
          $y//author/text()
        else
          "Inconnu"
      }
      ,
      element {"dt"}
      {
        "Titre"
      }
      ,
      element {"dd"}
      {
        $y/title/text()
      }
      ,
      element {"dt"}
      {
        "Sous-titre"
      }
      ,
      element {"dd"}
      {
        if($y/subtitle/text()) then
          $y/subtitle/text()
        else
          ()
      }
      ,
      element {"dt"}
      {
        "Date"
      }
    }
  )

```

```

    }
    ,
    element {"dd"}
    {
        $y/date/text()
    }
    ,
    element {"dt"}
    {
        "Lieu"
    }
    ,
    element {"dd"}
    {
        concat("Latitude : ", $y//lat)
        ,
        concat("Longitude : ", $y//lng)
    }
    ,
    element {"br"}
    {
        }
    }
)
}
}
}
}

```

#### **4 - Propriétés des requêtes XPath et XQuery**

-- //b[parent::a]

Reformulation :

/descendant-or-self::a/child::b

1.

<?xml version="1.0"?>

<a>

<b/>

</a>

2.

Cette requête est vide pour tous les documents xml valide par rapport à la DTD de l'énoncé.

En effet un élément a a forcément comme fils un noeud c, un noeud b ne peut donc pas avoir de parent

a. De plus l'élément b doit être déclaré dans la DTD.

-- //a/preceding-sibling::c



Reformulation :

/descendant::c[following-sibling::a]

1.

```
<?xml version="1.0"?>
```

```
<a>
```

```
  <c>
```

```
    <c/>
```

```
    <a>
```

```
      <c/>
```

```
    </a>
```

```
  </c>
```

```
</a>
```

2.

Cette requête est non-vide pour le document xml précédent qui est valide par rapport à la DTD de l'énoncé.

-- //c[preceding::d]

Reformulation :

/descendant::d/following::c

1.

```
<?xml version="1.0"?>
```

```
<a>
```

```
  <c>
```

```
    <d/>
```

```
    <a>
```

```
      <c/>
```

```
    </a>
```

```
  </c>
```

```
</a>
```

2.

Cette requête est vide pour tout document xml valide car l'élément d n'est pas déclaré dans le DTD.

-- //b/a/preceding-sibling::c/preceding::d

Reformulation :

d doit avoir following::c[(parent::b)and(following-sibling::a)]  
/descendant::d[following::c[(parent::b)and(following-sibling::a)]]

1.

```
<?xml version="1.0"?>
```

```
<a>
```

```
    <c>
```

```
        <d/>
```

```
        <b>
```

```
            <c/>
```

```
            <a>
```

```
                <c/>
```

```
            </a>
```

```
        </b>
```

```
    </c>
```

```
</a>
```

2.

Cette requête est vide pour tout document xml valide par rapport au DTD de l'énoncé car les éléments d et b ne sont pas décrits dans le DTD.

```
-- /a/b/../*/.../preceding::d
```

Reformulation :

/descendant-or-self::d[(following::node()/child::node())and(/a/b)]

1.

```
<?xml version="1.0"?>
```

```
<a>
```

```
    <c>
```

```
        <d/>
```

```
    </c>
```

```
    <b>
```

```
        <c/>
```

```
    </b>
```

```
</a>
```

2. Cette requête est vide pour un document xml valide par rapport au DTD car l'élément a ne peut contenir qu'un fils c et car les éléments b et d ne sont pas décrits dans le DTD.

```
-- //a/ancestor::b/parent::c/child::d/parent::e
```

Reformulation :

/descendant-or-self::e[(child::b/descendant::a)and(self::c/child::d)]

1.

Impossible de trouver un document xml pour lequel la réponse à la requête n'est pas vide car un noeud ne peut être à la fois de type e et de type c.

2.

Cette requête est vide pour n'importe quel document xml.

3. Pour les langages XPath et XQuery la propriété suivante est fausse :

Si  $X=Y$  et  $Y=Z$  alors  $X=Z$

Par exemple prenons le cas suivant :

```
<?xml version="1.0">
<a>
  <b>
    <e>Fraise</e>
    <e>Banane</e>
  </b>
  <c>
    <e>Banane</e>
    <e>Ananas</e>
  </c>
  <d>
    <e>Ananas</e>
    <e>Pamplemousse</e>
  </d>
</a>
```

On a  $/a/b = /a/c$  et  $/a/c = /a/d$  mais on n'a pas  $/a/b = /a/d$ .

4.

```
<?xml version="1.0"?>
<r>
  <a>Banane</a>
  <a>Banane</a>
</r>
```

5.  $/r[not(//a[not(@ref=//a/@id)])]$

/r[count(a)=count(//a[@ref=//a/@id])]

/r[(//a[@ref=//a/@id])[position()=count(//a)]]